What is claimed is:

1. A system for managing licenses for protected software on a communication network, the system comprising:

at least one client computer capable of being coupled to the communication network for requesting a commuter authorization to use the protected software and for storing a commuter authorization lifetime representing a time period for which the commuter authorization is valid; and

at least one license server coupled to the communication network, each license server programmed for managing a distribution of allocations to use the protected software and at least one license server programmed for granting a commuter authorization in response to a request for a commuter authorization;

wherein after a commuter authorization is communicated from a granting license server to a requesting client computer, the requesting client computer may use the protected software while coupled to or decoupled from the communication network until the commuter authorization lifetime expires.

2. A system as recited in claim 1, wherein while the requesting client computer maintains a valid commuter authorization, the requesting client computer may open the protected software multiple times, including simultaneous instantiations of the protected software.

3. A system as recited in claim 1, the at least one license server further programmed for granting a commuter authorization to the requesting client computer and decrementing a count of available allocations only if there is an available allocation in the at least one license server.

4. A system as recited in claim 3, the requesting client computer further including memory for storing commuter authorization information including the commuter authorization lifetime and a check-in value received from the granting license server when the granting license server grants the commuter authorization to the requesting client computer; and

the granting license server further including memory for storing commuter authorization information including the commuter authorization lifetime and a check-in value when the granting license server grants the commuter authorization to the requesting client computer.

5. A system as recited in claim 4, the requesting client computer programmed for returning the commuter authorization by setting its check-in value to a returned state and communicating a check-in message to the granting license server; and

the granting license server further programmed for setting its check-in value to the returned state and incrementing its count of available allocations upon receipt of the check-in message.

6. A system as recited in claim 5, wherein if the commuter authorization is not returned prior to an expiration of the commuter authorization lifetime, at the expiration of the commuter authorization lifetime:

the requesting client computer is further programmed for setting its check-in value to a returned state; and

the granting license server is further programmed for setting its check-in value to the returned state and incrementing its count of available allocations.

7. A system as recited in claim 1, the requesting client computer programmed for enabling a user to select the commuter authorization lifetime.

8. A system as recited in claim 1, the requesting client computer programmed for enabling a user to select the license server from which to request a commuter authorization.

9. A system as recited in claim 4, the at least one license server comprising a pool of license servers, and

the granting license server further programmed for communicating the commuter authorization lifetime and the check-in value stored in the granting license server to other license servers in the pool when the granting license server grants the commuter authorization to the requesting client computer, so that even if the granting license server should go down, another license server in the pool can act as the granting license server.

10. A system as recited in claim 1, the requesting client computer further programmed for detecting attempts to tamper with its internal clock and invalidating the commuter authorization if tampering is detected.

11. A method for managing licenses for protected software on a communication network, the method comprising the steps of:

coupling at least one client computer and at least one license server to the communication network;

communicating a request for a commuter authorization to use the protected software from the at least one client computer to the at least one license server over the communication network;

granting a commuter authorization to the at least one client computer from the at least one license server and decrementing a count of available allocations within the at least one license server only if there is an available allocation in the at least one license server; and

storing a commuter authorization lifetime representing a time period for which the commuter authorization is valid within the at least one client computer;

wherein after a commuter authorization is communicated from a granting license server to a requesting client computer, the requesting client computer may use the protected software while coupled to or decoupled from the communication network until the commuter authorization lifetime expires.

12. A method as recited in claim 11, wherein while the requesting client computer maintains a valid commuter authorization, the requesting client computer may open the protected software multiple times, including simultaneous instantiations of the protected software.

13. A method as recited in claim 11, the step of granting a commuter authorization to the at least one client computer from the at least one license server further including the steps of:

communicating the commuter authorization lifetime and a check-in value from the granting license server to the requesting client computer; and

storing the commuter authorization lifetime and the check-in value in the requesting client computer and the granting license server.

14. A method as recited in claim 13, wherein when the protected software is no longer needed, the method further includes the steps of:

returning the commuter authorization by setting the check-in value stored in the requesting client computer to a returned state and communicating a check-in message to the granting license server; and

setting the check-in value stored in the granting license server to the returned state and incrementing the count of available allocations stored in the granting license server when the granting license server receives the check-in message.

15. A method as recited in claim 14, wherein if the commuter authorization is not returned prior to an expiration of the commuter authorization lifetime, at the expiration of the commuter authorization lifetime the method further includes the steps of:

setting the check-in value stored in the requesting client computer to a returned state; and

setting the check-in value stored in the granting license server to the returned state and incrementing the count of available allocations stored in the granting license server.

16. A method as recited in claim 11, the step of communicating a request for a commuter authorization to use the protected software further including the step of selecting the commuter authorization lifetime.

17. A method as recited in claim 11, the step of communicating a request for a commuter authorization to use the protected software further including the step of selecting the license server from which to request a commuter authorization.

18. A method as recited in claim 13, the at least one license server comprising a pool of license servers, the method further including the steps of:

communicating the commuter authorization lifetime and the check-in value stored in the granting license server to other license servers in the pool when the granting license server grants the commuter authorization to the requesting client computer, so that even if the granting license server should go down, another license server in the pool can act as the granting license server.

19. A method as recited in claim 11, further including the step of detecting attempts to tamper with an internal clock of the requesting client computer and invalidating the commuter authorization if tampering is detected.